REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. AGENCY USE ONLY *(Leave blank)*	2. REPORT DATE	3. REPORT TYPE AND DATES COVERED
	6 May 97	

4. TITLE AND SUBTITLE
A Temporal Cascade approach for staircase Liner Programs wirh an Application to Air force Mobility Optimization

5. FUNDING NUMBERS

6. AUTHOR(S)
Steven F. Baker

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
Naval Postgraduate School

8. PERFORMING ORGANIZATION REPORT NUMBER

97-010D

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)
DEPARTMENT OF THE AIR FORCE
AFIT/CI
2950 P STREET
WRIGHT-PATTERSON AFB 45433-7765

10. SPONSORING/MONITORING AGENCY REPORT NUMBER

11. SUPPLEMENTARY NOTES

12a. DISTRIBUTION AVAILABILITY STATEMENT

DISTRIBUTION STATEMENT A
Approved for public release;
Distribution Unlimited

12b. DISTRIBUTION CODE

13. ABSTRACT *(Maximum 200 words)*

19970512 151

14. SUBJECT TERMS

15. NUMBER OF PAGES
54

16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT	18. SECURITY CLASSIFICATION OF THIS PAGE	19. SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT

DTIC QUALITY INSPECTED 3

A Temporal Cascade Approach for Staircase Linear Programs with an Application to Air Force Mobility Optimization

Dissertation Proposal by Steven F. Baker

Operations Research Department,

Naval Postgraduate School

May 22, 1996

Abstract

This research outlines a method by which a staircase linear program that optimizes decisions over a finite time horizon can be approximated and bounded. A feasible solution is derived by a *Temporal Cascade Heuristic,* which sequentially solves overlapping subsets of the model's time periods. In turn, that approximation is bounded by a *Lagrangian Cascade,* which penalizes infeasibility by incorporating dual information provided by the heuristic's solution. A large temporal LP developed for USAF mobility planners provides the case study for the method's development. Early results from the Temporal Cascade Heuristic show the feasible solution to be of good quality, although the Lagrangian Cascade bounding scheme has not yet been implemented.

INTRODUCTION

Large linear programs (LP's) often require indirect solution methods that exploit the problem's structure. Models that incorporate time frequently contain such a structure, commonly known as a staircase. A staircase structure is formed when a linear program contains variables and constraints that *directly* affect only nearby time periods. This is easy to grasp intuitively in a scheduling situation; a dispatcher's decisions on the first day of the month will not have a major effect on the vehicle fleet by the 30th day, assuming the round trip times are only a day or so. The purpose of this research is to formalize one of the heuristics currently used to approximate the solution of a staircase LP, and provide a bound for that heuristic's accuracy.

The success of LP in aiding schedulers is well known. It offers the ability to enlarge the planning horizon (the number of time periods considered by the scheduler), as well as methodically compute the best option within a horizon. Unfortunately, LP's are limited by temporal considerations in at least two ways: 1) future uncertainty makes gathering accurate data for the latter periods of a planning horizon problematic; and 2) a sufficiently large planning horizon may produce a model which is too large to solve with current technology. A human scheduler faces the same difficulties, namely reconciling the *increasing* number of options with *decreasing* certainty as the planning horizon grows. For either scheduler or LP, perhaps the most straightforward way of dealing with the difficulties incurred by a large planning horizon is to focus sequentially on a subset of the planning horizon's periods, then move forward in time to a new subset. This temporal "myopia" degrades the solution quality, but makes the problem simple enough to solve. Moreover, a model which is used to *mimic* scheduling, but not produce schedules, is best if it can incorporate the realism of nearsighted scheduling. For example, when choosing fleetsize or infrastructure for use in future dispatching, an LP wishes to optimize given the *current scheduling capabilities*, in-

2

stead of a utopian capability. Nonetheless, knowledge of a truly optimal solution is insightful, so any heuristic which degrades solution quality should be supplemented by an optimistic bound, in order to quantify myopia's cost.

As shown by the dispatcher problem, scheduling myopia is acceptable (and sometimes desirable for model realism), provided the commitments initiated by decisions are short relative to the myopia. Once a schedule is produced for a limited number of periods, the process can be cascaded forward in order to solve for a later set of periods. Mathematically, this implies generating a feasible solution by successively solving for only a subset of rows and columns, then moving to a set of rows and columns corresponding to later time periods. Each of these subproblems should overlap the previously solved subproblem in order to minimize the end effects caused by the temporal limitation. Fortunately, this methodology is facilitated by the structure of a staircase LP. The linkage of a time period's variables and constraints only to nearby periods manifests itself as an overlapping "staircase" along the main diagonal of the constraint coefficient matrix. The width of the overlap gives the number of time periods directly affected by the decisions (variable levels) made in a given time period. The rest of the coefficient matrix is relatively sparse, since variables (columns) associated with the early time periods rarely appear in constraints (rows) corresponding to the later time periods. This well known methodology is known as either the rolling horizon, or temporal cascades heuristic. However, the heuristic is sparsely documented, and is theoretically incomplete, since no scheme to bound the solution quality has been offered.

The quality of the solution produced by the above scheme is dependent on many scenario specific factors, and cannot be stated theoretically for most problems. However, this research intends to develop an optimistic bound (lower bound for a minimization problem) by exploiting information derived from this heuristic solution. Since many temporal LP's have only *direct* variable linkage between adjacent (or nearby)

3

time periods, relaxing the rows associated with certain time periods can decouple an otherwise linked monolith. As with most decompositions however, the success of this scheme is dependent on the ability to compute accurate prices for resource consumption of the relaxed constraints. With such prices, a Lagrangian penalty can be applied to the subproblems, and a lower bound can be derived. Often price selection is computationally intensive, which makes Lagrangian methods undesirable. However, in this case, reasonable prices are readily available from the cascade heuristic described above. This proposal explores the use of temporal cascades on a model currently in use by the Pentagon's mobility planners, and provides a bounding methodology for the quality of the heuristic's solution.

LITERATURE REVIEW

The topics germane to the proposed research include decomposition of large LP's, cascade heuristics, Lagrangian relaxation, and military mobility optimization. While there is a wealth of literature on decomposition and Lagrangian relaxation, cascading and military mobility optimization are less well documented. Below is a summary of the literature.

The notion of incorporating dual information to decompose large linear programs into smaller, structured LP's originated with Dantzig & Wolfe (6), and Benders (4). With respect to transportation related problems, Geoffrion & Graves (7) used Benders' decomposition in a resource directed scheme to reduce a mixed integer, multicommodity flow problem into separable single commodity problems. Brown *et al* (8) extended this technique using elastic constraints to insure feasibility as well as speed convergence. Decomposition has also been applied to staircase linear programs by Glassey (10) as well as Ho and Manne (15). This method repeatedly applies the Dantzig-Wolfe technique to succeeding (or preceding) levels of a staircase LP, forming a "nested" decomposition. A staircase LP can also be decomposed by Benders'

4

method, as given by Van Slyke & Wets (27) for 2 stage stochastic programming, then later by Birge (5) in multistage stochastic programming. Finally, methods of advanced basis selection and preliminary cut generation for nested models are offered by Morton (21), who offers one of many applications to stochastic programming in the current literature.

Although not a decomposition technique, the solution of large scale LP's can also be accomplished by aggregation of time periods until developing a problem of workable size. Zipkin (31) describes a methodology for bounding the error incurred by such aggregation in some problems.

Lagrangian relaxation is widely used in many applications of optimization, including vehicle routing, travelling salesman, and network design problems (1). Common to these methods is a multiplier search, which has proved the most difficult aspect of the overall method. A summary of search techniques is given in Parker & Rardin (24), as well as Bazaraa *et al* (3). Subgradient search techniques are perhaps the most common, although ensuring movement along a good, or even improving subgradient is computationally expensive. Progress in this area was reported from Kim & Ahn (18), who modified the traditional method (given in Held *et al* (14)) by developing a convergence scheme based on a weighting of all previous iteration's subgradients. Hearn & Lawphongpanich (13) also reported an improved multiplier search, but took an outer linearization approach which included an aggregate of previous cuts in each iteration's cut set. Finally, both Han (12) and Tseng (25) developed multiplier search methods based on a successive projection algorithm, which employs proximal penalty terms to drive the convergence of linking primal or dual variables within subproblems that are otherwise separable.

The use of temporally progressing heuristics in optimization is of two varieties; cascading and forward optimization. Cascading, or successively solving only a portion of the time periods in order to produce an advanced basis was used and reported by

Brown *et al* (9). Jayakumar & Ramasesh (16) analyzed the computational savings of this technique on a number of test problems. Forward optimization as outlined by Morton (23) involves solving successively longer (more time periods) problems until a *decision horizon* is reached. A decision horizon is a point beyond which solving larger problems will not alter the decisions of the first time period. This method shows that (for some problems) an optimal solution can be reached by solving a succession of small LP's, and recording the values of the first time period within each as optimal. Aronson *et al* (2) develop and test this idea for certain classes of problems, notably from the area of production scheduling and inventory control. Related work is done by Manne (20), who provides sufficient conditions for optimality when truncating infinite horizon LP's whose coefficients do not change in the latter periods. Walker (28) extends this idea to bound the error produced by truncating infinite horizon LP's prior to Manne's criteria. Unfortunately, the forward and infinite horizon methods require either an unchanging (homogeneous) or some other special structure, which does not exist for many staircase problems. There is not a body of literature on the solution and bounding of large, but still finite LP's by a *temporal cascade heuristic*-which successively solves a portion of a nonhomogeneos staircase LP in order to approximate an otherwise intractable problem.

Until recently, the computational demands of LP in modelling large scale military contingency deployments allowed an insufficient level of detail to be useful. As such, simulation was the method of choice for analyzing fleet mix and infrastructure requirements of such a deployment. Wing *et al* (29) developed an LP as a response to the Mobility Requirements Study mandated by the National Defense Authorization Act of 1991. Yost (30) continued the introduction of LP into the mobility modelling arena with the development of THRUPUT in 1994, which provided greater detail for the airlift aspect of the deployment scenario. Concurrent with this work, the RAND Corporation developed CONOP (17), which also focused on airlift, but initially was

used to examine the efficacy of aerial refueling of airlift aircraft in a contingency. Lim (19) extended THRUPUT with the development of THRUPUT II, which incorporated the multiple time periods of a contingency. Subsequently, Goggins (11) studied the effects of uncertainty on THRUPUT II, and Turker (26) examined the impact of airfield and route fidelity of the same model. Other THRUPUT II enhancements are ongoing, including the research described in this proposal.

CASE STUDY FOR CASCADE IMPLEMENTATION

Background

The airlift mobility model currently under development for the Air Force Studies and Analysis Agency (AFSAA) provides the need for the temporal cascades heuristic outlined above. Our first full scale runs produced model instances with nearly 3 million nonzeros. After some variable consolidation and fidelity coarsening, the same scenario was reduced to 1.7 million nonzeros. Unfortunately, this is still large given AFSAA's limited on-site computing. Fortunately THRUPUT II is a good candidate for the temporal cascades heuristic, as it fits the scheduler's paradigm with minimal alteration.

THRUPUT II Model Formulation

Prior to describing the proposed solution strategy in detail, below is the complete formulation of the Air Force mobility model, as given by Morton *et al* (22):

Indices

u Military units to be moved

a Aircraft types

t, t' Time periods

f All airfields (origins, enroutes and destinations)

i Origin airfields

k Destination airfields

r Routes

Index sets

F	Available airfields
$I \subseteq F$	Origin airfields
$K \subseteq F$	Destination airfields
A	Aircraft types
$A_{bulk} \subseteq A$	Bulk cargo capable aircraft
$A_{ovr} \subseteq A_{bulk}$	Oversize cargo capable aircraft
$A_{out} \subseteq A_{ovr}$	Outsize cargo capable aircraft
R	Available routes
$R_a \subseteq R$	Available routes for aircraft a
$R_{af} \subseteq R_a$	Available routes for a that use airfield f
$R_{aik} \subseteq R_a$	Available routes for a with origin i and destination k
$DR_i \subseteq R$	Delivery routes from origin i
$RR_k \subseteq R$	Recovery routes from destination k
T	Time periods
$T_{uar} \subseteq T$	Allowable launch times for u, a, r combination

Data

$movepax_{uik}$	Troop movement requirement of u from i to k
$moveue_{uik}$	Cargo movement requirement of u from i to k
$proover_u$	Proportion of u's cargo that is oversize
$proout_u$	Proportion of u's cargo that is outsize

$latepenue_u$	Daily lateness penalty per ton of u's cargo
$latepenpax_u$	Daily lateness penalty per u's pax
$nogopenue_u$	Non-delivery penalty per ton of u's cargo
$nogopenpax_u$	Non-delivery penalty per u's pax
$maxlate$	Maximum allowed lateness for any u
$dayslate_{uart}$	Days late of u if launched on t along r by a
$preserve_a$	Nominal cost (penalty) for aircraft usage
$uesqft_u$	Ft^2 per ton of u's cargo
$paxwt$	Passenger weight
$supply_{at}$	New aircraft supply available at time t
$cumsup_{at}$	Cumulative supply of new aircraft by time t
$maxpax_a$	Passenger capacity of aircraft a
$paxsqft_u$	Ft^2 per passenger
$acsqft_a$	Ft^2 available per aircraft a
$loadeff_a$	Proportion of available aircraft space
$urate_a$	Daily fraction of aircraft availability
$mogcap_{ft}$	Airfield capacity of narrow body aircraft
$mogreq_{af}$	Airfield space used by an aircraft in narrow body units
$mogeff_f$	Airfield space utilization efficiency
$maxload_{ar}$	Max payload of a on route r
$gtime_{afr}$	Ground time of a at f flying route r
$rctime_{afr}$	Rounded cumulative time to f by a flying on r
$flttime_{ar}$	Flight time of a flying on r
$ctimep_{ar}$	Cumulative time to offload by a flying r
$rctimep_{ar}$	Rounded cumulative time to offload by a flying r
$mtime_{art't}$	Time flown in t of mission launched at t' by a along r
$maxflt$	Maximum mission duration

$$m \qquad maxflt - 1$$

Variables {ALL VARIABLES ARE NON-NEGATIVE}

X_{uart}	Missions launched at t for u flown by a along r
Y_{art}	Recovery missions launched at t flown by a along r
$ALLOT_{ait}$	New aircraft allotted to airfield i at t
$RELEASE_{ait}$	Surplus aircraft released from airfield i at t
H_{ait}	Aircraft held overnight at airfield i at t
HP_{akt}	Aircraft held overnight at airfield k at t
$NPLANES_{at}$	Aircraft in the airlift system at t
$CONSUME_{at}$	revised form of $NPLANES$ (discussed in text)
$TONSUE_{uart}$	Tons of u moved by a along r at t
$TPAX_{uart}$	Pax of u moved by a along r at t
$UENOGO_u$	Cargo of u not moved
$PAXNOGO_u$	Pax of u not moved

Formulation

$$Z = \min \sum_{\substack{u,a, \\ r \in R(a), \\ t \in T(uar)}} (latepenue_u \cdot dayslate_{uart} \cdot TONSUE_{uart} \tag{1}$$

$$+ latepenpax_u \cdot dayslate_{uart} \cdot TPAX_{uart})$$

$$+ \sum_u (nogopenue_u \cdot UENOGO_u + nogopenpax_u \cdot PAXNOGO_u)$$

$$+ \sum_{a,t} preserve_a \cdot NPLANES_{at}$$

OBJECTIVE: Minimize the sum of late and undelivered cargo, plus a nominal tie breaking penalty for aircraft used.

Subject To

$$\sum_{\substack{a\in A(bulk), \\ r\in R(aik), \\ t\in T(uar)}} TONSUE_{uart} + UENOGO_u = moveue_{uik}$$

(2)

$$\forall u, i, k : moveue_{uik} > 0$$

DELIVERY REQUIREMENTS: The sum of on-time, late, and undelivered cargo must equal the delivery requirements for each unit.

$$\sum_{\substack{a\in A(out), \\ r\in R(aik), \\ t\in T(uar)}} TONSUE_{uart} + UENOGO_u \geq proout_u \cdot moveue_{uik}$$

(3)

$$\forall u, i, k : moveue_{uik} > 0$$

OUTSIZE DELIVERY : At least the required portion of delivered cargo must be outsize.

$$\sum_{\substack{a\in A(ovr), \\ r\in R(aik), \\ t\in T(uar)}} TONSUE_{uart} + UENOGO_u \geq (proover_u + proout_u) \cdot moveue_{uik}$$

(4)

$$\forall u, i, k : moveue_{uik} > 0$$

OVERSIZE DELIVERY : At least the required portion of delivered cargo must be oversize.

$$\sum_{\substack{a, \\ r\in R(aik), \\ t\in T(uar)}} TPAX_{uart} + PAXNOGO_u = movepax_{uik}$$

(5)

$$\forall u, i, k : movepax_{uik} > 0$$

PAX DELIVERY REQUIREMENTS: The sum of on-time, late, and undelivered pax must equal the delivery requirements for each unit.

$$\sum_{u,r\in DR(i)} X_{uart} + H_{ait} + RELEASE_{ait}$$
$$= H_{ai,t-1} + ALLOT_{ait} + \sum_{\substack{r\in R(aik), \\ t'+rctimep(ar)=t}} Y_{art'} \qquad \forall a, i, t$$

(6)

11

ORIGIN FLOW BALANCE : The sum of departing plus layover aircraft must equal returning and newly available aircraft, plus layover aircraft from the previous period.

$$\sum_{r\in RR(k)} Y_{art} + HP_{akt} = HP_{ak,t-1} + \sum_{\substack{u,r\in R(aik),\\ t'\in T(uar),\\ t'+rctimep(ar)=t}} X_{uart'} \quad \forall a,k,t \tag{7}$$

DESTINATION FLOW BALANCE : Same as above without allotments and releases

$$\sum_{i} ALLOT_{ait} \le supply_{at} \quad \forall a,t \tag{8}$$

AIRCRAFT ALLOTMENT : Only supplied aircraft can be allotted..

$$NPLANES_{at} = \sum_{i,t'\le t} ALLOT_{ait'} - \sum_{i,t'\le t} RELEASE_{ait'} \quad \forall a,t \tag{9}$$

AIRCRAFT COUNT : Planes in the system equal allotments minus releases.

$$\sum_{\substack{r\in R(a),\\ t'\le t,u}} \tau_{artt'} X_{uart'} + \sum_{\substack{r\in R(a),\\ t'\le t}} \tau_{artt'} Y_{art'} + \sum_{i,t'\le t} H_{ait'} + \sum_{k,t'\le t} HP_{akt'}$$

$$\le \sum_{t'\le t} NPLANES_{at'} \quad \forall a,t \tag{10}$$

where

$$\tau_{artt'} = \left\{ \begin{array}{ll} t - t' + 1 & if\ t' \le t < t' + ctimep_{ar} - 1 \\ ctimep_{ar} & if\ t \ge t' + ctimep_{ar} - 1 \end{array} \right\}$$

AIRCRAFT CONSUMPTION : Reduce discretization effects in aircraft usage.

$$TONSUE_{uart} + paxwt \cdot TPAX_{uart} \le maxload_{ar} \cdot X_{uart} \quad \forall u,a,r,t : t \in T_{uar} \tag{11}$$

AIRCRAFT WEIGHT : Do not overload aircraft.

$$paxsqft_u \cdot TPAX_{uart} + uesqft_u \cdot TONSUE_{uart} \leq acsqft_a \cdot loadeff_a \cdot X_{uart} \tag{12}$$

$$\forall u,a,r,t : t \in T_{uar}$$

AIRCRAFT SPACE : Do not "overbulk" aircraft.

$$TPAX_{uart} \leq maxpax_a \cdot X_{uart} \quad \forall u,a,r,t : t \in T_{uar} \tag{13}$$

AIRCRAFT SEATS: Do not overfill aircraft with passengers

$$\sum_{\substack{u,a,r\in R(a), \\ t'\in T(uar), \\ t'+rctime(afr)=t}} mogreq_{af} \cdot gtime_{afr} \cdot X_{uart'} \tag{14}$$

$$+ \sum_{\substack{a,r\in R(a), \\ t'+rctime(afr)=t}} mogreq_{af} \cdot gtime_{afr} \cdot Y_{art'}$$

$$\leq mogeff_f \cdot mogcap_{ft} \qquad \forall f,t$$

AIRFIELD CAPACITY LIMITIATIONS: The number of aircraft using an airfield must be less than an airfield's capacity. It is enforced each period for each airfield.

$$\sum_{\substack{u,r\in R(a), \\ t\in T(uar)}} flttime_{ar} \cdot X_{uart} + \sum_{r\in R(a),t} flttime_{ar} \cdot Y_{art} \tag{15}$$

$$\leq \sum_t urate_a \cdot NPLANES_{at} \quad \forall a$$

AIRCRAFT UTILIZATION RATE: Over the course of the entire run, aircraft cannot be flown more than is historically reasonable from a maintenance standpoint. Enforced for each aircraft type.

In order to make the model more tractable and conducive to the temporal cascades heuristic, several enhancements are appropriate. Although not the focus of this research, the next section describes the modifications which facilitate the heuristic.

Model Enhancements and Modifications

In the midst of the AFSAA study, we decided to force aircraft allocations only on days where new supply became available. That allowed us to eliminate the relatively dense lower triangular structure of the constraint, and replace it with (8) above. We pay a minor objective function price for this; aircraft can no longer be allocated "just-in-time" for a peak movement requirement. This is a trivial concern given the nominal penalty on $NPLANES$. Furthermore, the change offers more than just reduction of (8)'s density; we can now eliminate the lower triangular structure of constraints (9) and (10). (9) now becomes a simple balance of flow constraint:

$$NPLANES_{a,t-1} + \sum_i ALLOT_{ait} - \sum_i RELEASE_{ait} = NPLANES_{a,t} \qquad \forall a, t$$

$$(16)$$

However, the alteration described below removes this constraint altogether.

As stated, (10) and (16) compute $NPLANES$, the number of aircraft in the system. The objective function assesses a nominal penalty against $NPLANES$, in order to favor a smaller fleet in case of tie. This is a sensible approach for civilian aircraft, whose release is a relatively permanent decision. However, it is not the most logical approach for military aircraft, which may provide discontinuous support for the deployment under study. Moreover, even a few days "rest" at home station allows for periodic maintenance completion, and should therefore not be counted against $NPLANES$ "in the system." In order to address this concern, define the variable $CONSUME_{at}$ to be the sum of enroute aircraft plus the sum of those incurring an in-theater holdover (HP). Embarkation holdovers (H) can be loosely defined as in the continental United States, and therefore home-station holdovers. Thus, H should not be included in $CONSUME$ (ideally, single period holdovers are probably not productive from a maintenance standpoint, but the oversight is small). This modification is made to (10), discussed shortly. Constraint (9) is replaced with a

14

new constraint which enforces the limit on total available aircraft, and consequently includes CONUS holdovers H :

$$CONSUME_{at} + \sum_i H_{ait} \leq cumsup_{at} \qquad \forall a, t \qquad (17)$$

So (10) now computes the number of aircraft supporting the scenario, and (17) will bound that number by the aircraft available. The $NPLANES$ variable is eliminated, and the objective function substitutes $CONSUME$ for it. Additionally, the allotment constraint (8) becomes an equality constraint, since all aircraft can be "inventoried" at a home station without a consumption penalty.

Originally, (10) was designed to reduce discretization effects resulting from the fact that travel times are usually are not integer. However, these effects only concern missions currently being flown, and need not be enforced from the first time period. Given $m = maxflt - 1$, note that missions launched as long ago as $t - m$ could be utilizing aircraft-days in time t. Summing all missions from $t - m$ to t includes all such missions. The number of aircraft days used by $X_{uart'}$ in time t is:

$$0 \text{ if } (t' + ctimep) \leq t$$

$$1 \text{ if } (t' + ctimep) \geq (t + 1)$$

$$ctimep - (t - t'), \text{ if } t < (t' + ctimep) < (t + 1)$$

This number is parameterized as $mtime_{a,r,t',t}$, the time incurred in period "t" by aircraft "a" launched along route "r" at time "t'". Note that no rounding occurs, so discretization effects are removed, just as they were in the previous formulation of (10). The revised constraint is:

$$\sum_{\substack{u,r, \\ (t-m) \leq t' \leq t}} mtime_{art't} \cdot X_{uart'} + \qquad (18)$$

$$\sum_{\substack{r, \\ (t-m) \leq t' \leq t}} mtime_{art't} \cdot Y_{art'} + \sum_{k} HP_{akt} \leq CONSUME_{at} \qquad \forall a, t$$

Since summing *mtime* over all time periods is equivalent to τ in equation (10), the new formulation is at least as strong. Moreover, the revision precludes "borrowing" of consumption from previous time periods– a problem encountered with the original formulation.

The motive for adjusting the aircraft consumption penalty is straightforward: the decision to release an aircraft is final, and therefore not conducive to a temporal cascades framework where the future is not known in all but the final cascade. The change is no more complex than the current formulation, and offers equal or greater realism.

Finally, the model should enforce aircraft utilization rate (UTE) more frequently. The entire scenario length is far too long to allow aircraft to operate continuously. A more practical limit is between two and three weeks. Again there is an ulterior motive for this, since the temporal cascades approach lends itself to breaks of approximately this length. As such, the revised model enforces UTE rate at the end of each cascade, described in greater detail later.

There is one additional major modification of the model undertaken prior to, but not in direct support of, the cascading methodology. To increase tractability, intra-theater airfields are frequently aggregated in order to reduce the number of routes generated. Unfortunately, this aggregation sharply reduces the fidelity of the model due to large disparities in the airfield capacities (known as Maximum on Ground, or MOG). This effect can be minimized by careful aggregation and set creation. In this scheme, first create subsets that contain units debarking at each theater airfield, say k_u. Define MOG constraints as usual for each theater airfield k, but define routes and the destination balance of flow constraint for *only one* of the airfields (preferably

one that is geographically near the middle). Next, sum the MOG constraints only over those X_{uart} variables that move units into airfield k, *not* all the routes associated with the "centroid" airfield. Finally, include all X_{uart} variables (associated with the theater) in the single flow balance constraint. Using this approach, balance of flow is aggregated but MOG is not, so fidelity is retained without the explosion of route cardinality associated with multiple destinations. Of course, the technique sacrifices a minor amount of geographical fidelity, since route times to a centroid will vary slightly from the true time to destination. Additionally, the inclusion of Y_{art} variables in MOG constraints is lost, since those variables are not tied to a unit. Fortunately, most situations do not require return routing through other destinations, so the loss of accuracy in assigning "Y" MOG is minimal. Implementation of the centroid scheme verified this; solution time was reduced and fidelity of intra-theater airfield capacity increased markedly.

In summary, the model now consists of the objective and constraints (2-8), (11-15), (17), and (18). Additionally, (8) becomes an equality constraint, constraint (6) no longer includes the "*RELEASE*" term, and (1) and (15) substitute "*CONSUME*" for "*NPLANES.* " The above changes increase model tractability, and allow for a smooth transition to temporal cascades. They are used throughout the remainder of the paper.

GENERIC CASCADES MODEL FORMULATION

As seen above, the full structure of THRUPUT II is complex. However, for the purpose of describing the solution methodology, it can be simplified into a very few types of structural constraints. The demand satisfaction constraints (2, 3, 4, 5) typically span between 10 and 30 days. They are labeled "extended staircase" constraints, because when viewed in tableau form, they form a thick staircase band along the main diagonal. Additionally, the extended staircase constraints are characterized by

their elastic penalties. Many of the other constraints have a staircase structure, but only span 2 or 3 periods; hence they are called staircase constraints (6, 7, 14, 18). Still other constraints span only a single time period, and are the easiest to deal with in a temporal setting (8, 11, 12, 13, 17). Finally, there are "block" constraints, which span every time period of a three week interval, but do not overlap (15).

In order to streamline the explanation of the temporal solution schemes, below is a greatly simplified version of the model. Gone are multiple aircraft types and routes, and infrastructure resource constraints (aircraft usage and MOG) are generic.

Indices

$u \in U$ Units

$t \in T$ Time

$UT_u \subset T$ Allowable movement times for u

$B \subset T$ Blocks of time (utilization rate enforcement)

Data

d_u Demand for unit u

s_t Resource supply (e.g. planes, MOG)

f_u, a_u Utilization factors (e.g. mission time)

$g(s_t), h(\sum s_t)$ Functions of s_t (e.g. space available)

$maxflt$ Mission duration (defined as a constant generically)

m $maxflt - 1$

Variables

$X_{u,t}$ Missions of unit u flown at time t. $X_{u,t}$ is only defined for $t \in UT_u$

P_u Nondelivered units of u

Formulation

$$Z = \min \sum_u P_u$$

$$\text{ST} \quad \sum_t X_{ut} + P_u \geq d_u \quad \forall u \qquad \text{Extended Staircase}$$

$$\sum_u f_u \cdot X_{u,t-m} + \sum_u f_u \cdot X_{ut} \le s_t \qquad \forall t \qquad \text{Staircase}$$

$$\sum_u a_u \cdot X_{ut} \le g(s_t) \qquad \forall t \qquad \text{Single Period}$$

$$\sum_{u,t \in B} X_{ut} \le h(\sum s_t) \qquad \forall B \qquad \text{Block}$$

The above formulation has a structure which can be exploited using the strategies described in the remainder of this paper. That structure is best illustrated by the following tableau with $m = 1, \mid U \mid = 4, \mid T \mid = 5$:
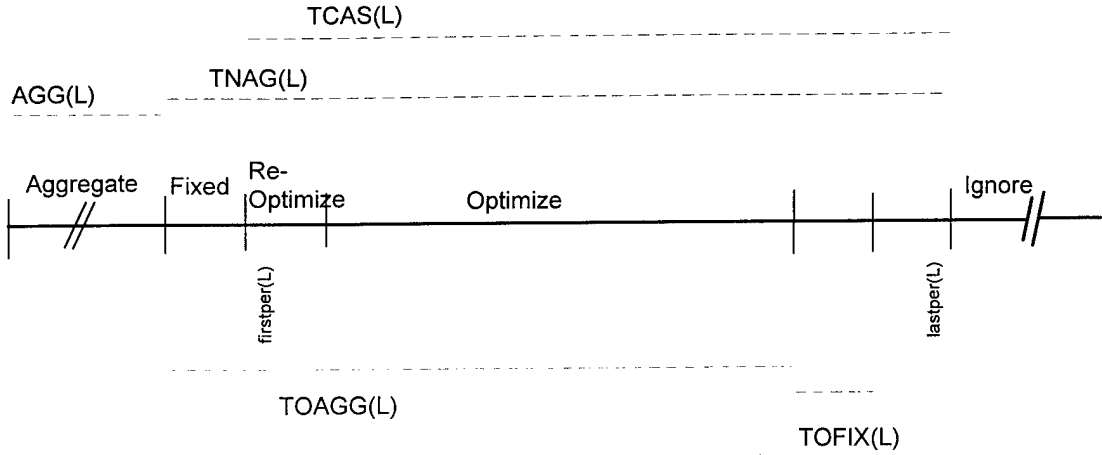
	X11	X21	X12	X22	X32	X13	X23	X33	X43	X24	X34	X44	X25	X45	P1	P2	P3	P4	RHS	
Obj															x	x	x	x		
u1	x		x			x									x				> d1	
u2		x		x			x			x			x			x			> d2	Extended
u3					x			x			x						x		> d3	Staircase
u4									x			x		x				x	> d4	
t1	x	x																	< s1	
t2	x	x	x	x	x														< s2	
t3			x	x	x	x	x	x	x										< s3	Staircase
t4						x	x	x	x	x	x	x							< s4	
t5										x	x	x	x	x					< s5	
t1	x	x																	< g(s1)	
t2			x	x	x														< g(s2)	Single
t3						x	x	x	x										< g(s3)	Period
t4										x	x	x							< g(s4)	
t5													x	x					< g(s5)	
b1	x	x	x	x	x														< h(s1)	Block
b2						x	x	x	x	x	x	x	x	x					< h(s2)	

TEMPORAL CASCADES UPPER BOUND

The combination of constraint elasticity and staircase structure allow a feasible solution of THRUPUT II by temporal cascades yielding an upper bound. This section outlines this decomposition, first by describing the heuristic generically, then by giving the specific implementation for the Air Force model. The section ends with a discussion and verification of solution feasibility..

As stated above, the characteristics of this model make it very conducive to temporal cascades. The "staircase" constraints contain variables which span a maximum of *maxflt* periods, (usually about 3 days in THRUPUT II, but shown as 2 in the tableau above). This limited constraint span with respect to time is key to the facilitation of feasibility in a suitably implemented temporal cascading scheme. The "extended staircase" constraints span many more periods, but will not violate feasibility because they are elastic. The single period constraints pose no difficulty, as they are completely separable. The "block" constraint is patterned after THRUPUT II's aircraft utilization constraint, which originally spanned the entire scenario length, but (as previously discussed) is only enforced with modeler discretion at regular intervals. Were it not for flexibility of this constraint, the proposed method would not work, so a partitioning of any constraint which spans the entire scenario is critical.

Below is the revised formulation which implements the temporal cascades upper bound:

TCAS(L)

AGG(L)   TNAG(L)

Aggregate | Fixed | Re-Optimize | Optimize | | | Ignore

firstper(L)

lastper(L)

TOAGG(L)

TOFIX(L)

Temporal Cascades Notation.—

L Cascade Index

$firstper_L$ First period in cascade L

$lastper_L$ Last period in cascade L

$TCAS_L = \{t : firstper_L \leq t \leq lastper_L\}$ Time periods in the current cascade

$TNAG_L = \{t : firstper_L - m \leq t \leq lastper_L\}$ The current unaggregated periods

$TOAGG_L = \{t : firstper_L - m \leq t < firstper_L - 2m\}$ The periods to be aggregated at the end of the current cascade.

$TOFIX_L = \{t :\in TNAG_L, t \notin TCAS_L\}$ The periods to be fixed at the end of the current cascade

$AGG_L = \{t : t < firstper_{L-1} - m\}$ The periods previously aggregated

$requag_{u,L}$: Aggregated amount of unit u moved from $t \in AGG$

Summary of the Temporal Cascades Heuristic.—

0) Set L=1, establish the size of each cascade $| TCAS |$. Also set $firstper_1 = 1$,

21

$lastper_1 = | TCAS |$. Establish other sets as described above.

1) Solve for variables and constraints in the current cascade window $(TCAS)$.

2) Set $requag_{u,L} = \sum_{t \in TOAGG(L)} X_{u,t} + requag_{L-1}$ for all units. $TOAGG_L$ includes all time periods not previously aggregated, but prior to $t \in TOFIX_L$

3) Fix variable levels for all $t \in TOFIX_L$. This ensures that staircase constraints associated with the early periods of the next cascade accurately reflect previously committed resources.

4) If $lastper_L \geq | T |$, terminate and report current solution. Otherwise, set $firstper_{L+1} = firstper_L + | TCAS | - m$, and $lastper_{L+1} = lastper_L + | TCAS | - m$.

5) Set $L = L + 1$. Update sets $TCAS, TNAG, TOAGG, TOFIX, AGG$ as described.

6) Return to 1).

Temporal Cascades Formulation.—

$$\overline{Z}_L = \min \sum_u P_u$$

OBJECTIVE: Minimize the sum of nondelivery penalties. The overall solution is given by $\overline{Z} = \overline{Z}_{\max(L)}$, which is the objective function value of the final cascade in the iteration.

$$\text{ST} \qquad \sum_{t \in TNAG(L)} X_{ut} + P_u \geq d_u - requag_{u,L-1} \qquad \forall u \quad (\alpha_u)$$

EXTENDED STAIRCASE: Include only columns of $TNAG_L$, and reduce the RHS requirement by deliveries aggregated from the previous cascades. α_u represents the dual variable associated with this constraint.

$$\sum_u f_u \cdot X_{u,t-m} + \sum_u f_u \cdot X_{ut} \leq s_t \qquad \forall t \in TCAS_L \quad (\beta_t)$$

STAIRCASE: Enforce for $t \in TCAS_L$, thus a minimum of m periods prior to

$firstper_L$ must remain unaggregated for this cascade (although they will be fixed). β_t represents the dual variable associated with this constraint

$$\sum_u a_u \cdot X_{ut} \leq g(s_t) \qquad \forall t \in TCAS_L$$

SINGLE PERIOD: Same as basic formulation, but define only for $t \in TCAS_L$.

$$\sum_{u,t \in B} X_{ut} \leq h(\sum s_t) \qquad \forall B \subseteq TCAS_L$$

BLOCK: Enforce only the block constraints whose columns are subsets of $TCAS$. This assumes the block constraints can be partitioned in this manner.

Note that the staircase constraint specifies the *minimum* number of overlap periods as m. The overlap should rarely be set at this minimum, since solution quality will suffer considerably at the minimum value. However, the minimum value ensures that model variables will only be fixed when all constraints that they appear in have been feasibly generated once. Given a sufficiently small feasible region, this might not be enough to ensure feasibility of subsequent constraints (pathologically, the feasible region could consist of only 1 point, which might not be found without solving the monolith). Fortunately, the structure of THRUPUT II precludes this, and is discussed later. However, it appears that defining an overlap to ensure feasibility is rather model specific, and cannot be stated generally.

Aside from the issue of feasibility, overlap size plays a role in the heuristic's solution quality. Large overlaps provide a smoother transition between cascades, and result in a closer approximation of the optimal solution. Intuitively this is clear; a scheduler can do better if presented with updated requirements at more frequent intervals. Thus the overlap m should be established as a compromise between solution time and solution quality.

A cascade consists of running the model for only a subset of the scenario's time periods. An iteration is complete when all time periods have been addressed by a

cascade. A unit's delivery requirement (extended staircase constraint) is not included in the iteration until its delivery window overlaps a time period of a cascade. Still, early cascades are likely to incur large penalty costs, since units may have only a few delivery periods available prior to the end of the cascade. Preliminary experiments reduced the nondelivery penalty for these units (by an amount proportional to the reduction of the units' delivery window), although further research will determine the efficacy of this technique. To date, the temporal cascades method works well when applied to the THRUPUT II model, producing feasible solutions with objective values only a few percent over the known optimal when the overlap (m) is large. Below is an in depth description of the THRUPUT II implementation.

THRUPUT II Specific Upper bound

Temporal Cascades Formulation.—

Below is the THRUPUT II formulation altered to implement the temporal cascades heuristic. The index sets are omitted for brevity except as they apply to the cascades.

$$\overline{Z}_L = \min \sum_{\substack{u,a,r, \\ t \in TNAG(L)}} (latepenue_u \cdot dayslate_{uart} \cdot TONSUE_{uart} \tag{19}$$

$$+ latepenpax_u \cdot dayslate_{uart} \cdot TPAX_{uart})$$

$$+ \sum_u (nogopenue_u \cdot UENOGO_u + nogopenpax_u \cdot PAXNOGO_u)$$

$$+ \sum_{a,t \in TNAG(L)} preserve_a \cdot CONSUME_{at} + agglate + aggcon$$

$$\text{where} \quad agglate = \sum_{\substack{u,a,r, \\ t \in AGG(L)}} (latepenue_u \cdot dayslate_{uart} \cdot TONSUE_{uart}$$

$$+ latepenpax_u \cdot dayslate_{uart} \cdot TPAX_{uart})$$

$$\text{and} \quad aggcon = \sum_{a,t \in AGG(L)} preserve_a \cdot CONSUME_{at}$$

24

$OBJECTIVE$: Similar to (1), but terms are delineated between aggregated and unaggregated time periods. *Agglate* and *aggcon* are the aggregated late and consumption penalties from earlier cascades. As before, the overall solution is given by $\overline{Z} = \overline{Z}_{\max(L)}$, which is the objective function value of the final cascade in the iteration.

Subject To (for each cascade L)

$$\sum_{\substack{a \in A(bulk), \\ r,t \in TNAG(L)}} TONSUE_{uart} + UENOGO_u = moveue_{uik} - requag_{u,L-1} \qquad (20)$$

$$\forall u, i, k : moveue_{uik} > 0$$

$$\text{where} \quad requag_{u,L-1} = \sum_{\substack{a \in A(bulk), \\ r,t \in AGG(L)}} TONSUE_{uart}$$

$DELIVERY \ REQUIREMENTS$: Original formulation of (2), but altered for equipment delivered in previous cascades.

$$\sum_{\substack{a \in A(out), \\ r,t \in TNAG(L)}} TONSUE_{uart} + UENOGO_u \geq proout_u \cdot moveue_{uik} - oureqag_{u,L-1}$$

$$\forall u, i, k : moveue_{uik} > 0$$
$$(21)$$

$$\text{where} \quad oureqag_{u,L-1} = \sum_{\substack{a \in A(out), \\ r,t \in AGG(L)}} TONSUE_{uart}$$

$OUTSIZE \ DELIVERY$: Original formulation of (3), but altered for previous cascades' outsize delivery.

$$\sum_{\substack{a \in A(ovr), \\ r,t \in TNAG(L)}} TONSUE_{uart} + UENOGO_u \geq$$

$$(proover_u + proout_u) \cdot moveue_{uik} - ovreqag_{u,L-1} \qquad \forall u, i, k : moveue_{uik} > 0$$
$$(22)$$

25

$$\text{where} \quad ovreqag_{u,L-1} = \sum_{\substack{a \in A(ovr), \\ r,t \in AGG(L)}} TONSUE_{uart}$$

OVERSIZE DELIVERY : Original formulation of (4), but altered for previous cascades' oversize delivery.

$$\sum_{\substack{a,r, \\ t \in TNAG(L)}} TPAX_{uart} + PAXNOGO_u = movepax_{uik} - preqag_{u,L-1} \tag{23}$$

$$\forall u, i, k : movepax_{uik} > 0$$

$$\text{where} \quad preqag_{u,L-1} = \sum_{q,r,t \in AGG(L)} TPAX_{uart}$$

PAX DELIVERY REQUIREMENTS: Original formulation of (5), but altered for previous cascades' pax delivery.

$$\sum_{u,r} X_{uart} + H_{ait} = H_{ai,t-1} + ALLOT_{ait} + \sum_{\substack{r, \\ t'+rctimep(ar)=t}} Y_{art'} \qquad \forall a, i, t \in TCAS_L \tag{24}$$

ORIGIN FLOW BALANCE : Original formulation of (6), but defined only for current cascade.

$$\sum_{r} Y_{art} + HP_{akt} = HP_{ak,t-1} + \sum_{\substack{u,r, \\ t'+rctimep(ar)=t}} X_{uart'} \quad \forall a, k, t \in TCAS_L \tag{25}$$

DESTINATION FLOW BALANCE : Original formulation of (7), but defined only for current cascade.

$$\sum_{i} ALLOT_{ait} = supply_{at} \quad \forall a, t \in TCAS_L \tag{26}$$

AIRCRAFT ALLOTMENT : Original formulation of (8), but defined only for current cascade, and set to equality (as previously addressed).

$$CONSUME_{at} + \sum_i H_{ait} \leq cumsup_{at} \qquad \forall a, t \in TCAS_L \qquad (27)$$

MAXIMUM PLANES : Equation (17), defined only for current cascade.

$$\sum_{\substack{u,r, \\ (t-m) \leq t' \leq t}} mtime_{art't} \cdot X_{uart'} + \qquad (28)$$

$$\sum_{\substack{r, \\ (t-m) \leq t' \leq t}} mtime_{art't} \cdot Y_{art'} + \sum_k HP_{akt} \leq CONSUME_{at} \qquad \forall a, t \in TCAS_L$$

PLANES CONSUMED : Equation (18), defined only for current cascade.

$$TONSUE_{uart} + paxwt \cdot TPAX_{uart} \leq maxload_{ar} \cdot X_{uart} \quad \forall u, a, r, t \in TCAS_L \quad (29)$$

AIRCRAFT WEIGHT : Original formulation of (11), but defined only for current cascade.

$$paxsqft_u \cdot TPAX_{uart} + uesqft_u \cdot TONSUE_{uart} \leq acsqft_a \cdot loadeff_a \cdot X_{uart}$$
$$\forall u, a, r, t \in TCAS_L \qquad (30)$$

AIRCRAFT SPACE : Original formulation of (12), but defined only for current cascade.

$$TPAX_{uart} \leq maxpax_a \cdot X_{uart} \quad \forall u, a, r, t \in TCAS_L \qquad (31)$$

AIRCRAFT SEATS: Original formulation of (13), but defined only for current cascade.

$$\sum_{\substack{u,a,r, \\ t'+rctime(afr)=t}} mogreq_{af} \cdot gtime_{afr} \cdot X_{uart'} \qquad (32)$$

27

$$+ \sum_{\substack{a,r, \\ t'+rctime(afr)=t}} mogreq_{af} \cdot gtime_{afr} \cdot Y_{art'}$$

$$\leq mogeff_f \cdot mogcap_{ft} \qquad \forall f, t \in TCAS_L$$

AIRFIELD CAPACITY LIMITIATIONS: Original formulation of (14), but defined only for current cascade.

$$\sum_{\substack{u,r, \\ (t-m)\leq t'\leq t, \\ firstper(L)\leq t<firstper(L+1)}} flttim_{art't} \cdot X_{uart'} + \sum_{\substack{r, \\ (t-m)\leq t'\leq t, \\ firstper(L)\leq t<firstper(L+1)}} flttim_{art't} \cdot Y_{art'}$$

$$(33)$$

$$\leq \sum_{\substack{firstper(L)\leq t<firstper(L+1)}} urate_a \cdot CONSUME_{at} \qquad \forall a$$

where $flttim_{art't}$ is defined as $mtime_{art't}$, minus any ground time spent during t' minus any ground time spent during $\{t', t\}$. This is a slightly more complex definition than $flttime_{ar}$, to account for block overlap.

AIRCRAFT UTILIZATION RATE: Enforce once per cascade, and include all flight time of the cascade, minus any overlap with the succeeding cascade.

Discussion.—

Consider a 90 period temporal cascades model with 25 periods in each cascade, and single day time steps. Given $maxflt = 3$, the minimum overlap (m) is 2. However, for greater solution quality, choose an overlap of 3. After optimal completion of the first cascade, the end effects must be considered. Conceivably, resources for the time periods just beyond the current cascade (26, 27 in this example, where $lastper = 25$) could have been committed in excess of their availability. MOG, for example, is committed up to 2 days in advance, since the longest mission time of this scenario is 3. However, MOG is constrained in the period in which it is *consumed*, not the

28

period in which it is *committed*. Since we have chosen to extend the overlap, the second cascade starts with $firstper = 23$. This permits reconsideration of period 23-25 decisions (permitting redress of MOG violations in periods 26,27), as well as a first look at MOG constraints in subsequent periods.

With a smooth transition between cascades ensured by overlapping the last three periods (23-25), decisions of periods 1-22 can be fixed. Variable levels in the first 19 of these periods are written to disk to conserve memory. Note that the resources consumed by period 1-19 decisions do not directly effect future cascade decisions, since resource commitments in these periods cannot reach period 23. However, unit cargo and pax delivered during the first 19 periods must be aggregated and subtracted from the requirements for those units (done in constraints (20) - (23)). Additionally, the penalties accrued for aircraft consumption and late cargo must be summed and appended to the next cascade's objective function. Nondelivery on the other hand, is not explicitly carried over, since all delivery constraints defined in the first cascade will be defined in subsequent cascades. Only the right hand sides of these constraints change to account for cargo delivered in the aggregated periods.

Finally, there are the periods just prior to $firstper$ of the next run (periods 20-22). These periods cannot yet be aggregated, since doing so would erase the fidelity of the resources committed in those periods. For example, a mission launched on period 22 will commit MOG in different periods than the same mission launched on period 21. Thus, the two must remain distinct. Aggregation of these periods will not come until after completion of the second cascade, since they do not have a direct impact on the third cascade.

An iteration terminates once all time periods have been addressed. The final objective function value is the upper bound for the entire problem. The solution is recorded piecemeal by saving the variable values of each run prior to aggregation.

"Null" Solution Feasibility The nature of the airlift problem ensures feasibility of the heuristic if $m \geq maxflt - 1$. To illustrate, suppose the heuristic has just solved for the periods of cascade L. Consider a null, or "state department" solution for cascade $L + 1$, i.e.: launch no more missions, and move no more cargo (X_{uart}, Y_{uart}, $TONSUE_{uart}$, $TPAX_{uart} = 0$ for $t \geq firstper_{L+1}$). Constraints (20-23) are elastic, and cannot be violated. Constraint (26) can force allocations arbitrarily, since the aircraft will be put in inventory at home station and consume no resources. Constraints (29-31) are of the form "$\leq$", have 0 on the left hand side using the null solution, and have a nonnegative right hand side. Thus they are not violated. Using the null solution, constraints (32) and (33) have temporally nonincreasing left hand sides, since returning missions cease to use MOG and flying hours. Therefore they remain feasible. Feasibility of (24), (25) is retained by holding as inventory (increasing variables H and HP) all aircraft as they complete their missions. (28) defines $CONSUME$, which is only used in the objective and in (27), and so cannot be infeasible by itself. The maximum planes constraint, (27) ensures the total aircraft on the ground plus those in the air do not exceed the supply. Since holding returning aircraft in inventory is a valid "use," an infeasibility here would portend a flaw in the basic (but not temporal cascades) formulation. Unfortunately, this is the case with the time discretization of THRUPUT II.

Time Discretized Feasible Solution In addition to other difficulties, time discretization effects can manifest themselves as "feasibility spoilers" in a temporal cascades formulation of THRUPUT II (specifically equation (27)). Under the "state department", or null solution, consider a mission fixed in cascade $L - 1$ that lands in cascade L. Further, suppose (27) is binding. The solver may be forced to immediately "launch" another mission of short duration; one that takes just less than 1 period. Otherwise, the aircraft is either held in inventory or flown on a longer mission, each

at a cost of 1 to the left hand side of (27). Thus a "fly" decision is feasible while a "wait" decision is not. Note this is a problem only when the mission duration is between half a time period and a full period. Interestingly, very short missions (less than 1/2 period) do not contribute to discretization infeasibility. These missions are rounded to zero, and will be forced by the balance of flow constraints to either be held in inventory or flown again. Each of these raises the LHS of (27), and would not occur in a fixed solution, where (27) is binding.

Showing feasibility of THRUPUT II under the temporal cascades scheme requires modification of the "null" solution. Infeasibility due to time discretization occurs when aircraft "savings" can be feigned by generating short duration "X" and "Y" missions, instead of full period "H" and "HP" holdovers. In the modified null solution, missions are flown (without cargo for simplicity) in cascade L if and only if: 1) they are in the (unfixed) overlap between $L-1$ and L; and 2) the mission duration $ctimep$ is less than a time period. Otherwise, the mission is cancelled and the aircraft held over. That way, the savings reaped by not flying aircraft meets or exceeds that incurred by increasing holdovers. Algebraically, we must demonstrate that the levels of H and HP forced in (24) and (25) do not contribute to an increase in the level of (27). First, define C as the set of cancelled missions:

$$C = \{X_{uart}, Y_{art} : t \in \{TCAS_{L-1} \cap TCAS_L\}, ctimep_{ar} \geq 1\}$$

$$\cup \{X_{uart}, Y_{art} : t \in TCAS_L, t > lastper_{L-1}\}$$

Plainly stated: cancel all but short missions during the overlap, and schedule no more thereafter. Assuming there are no additional allotments (which complicate the notation but do not alter the result), equation (24) becomes:

$$H_{ait} = H_{ai,t-1} - \sum_{u,r,} X_{uart} + \sum_{\substack{r, \\ t'+rctimep(ar)=t}} Y_{art'} \quad \forall a, i, t \in \{TCAS_{L-1} \cap TCAS_L\}$$

Which can be separated by the cancellation criterion:

$$= H_{ai,t-1} - \sum_{\substack{u,r, \\ X(uart)\notin C}} X_{uart} - \sum_{\substack{u,r, \\ X(uart)\in C}} X_{uart} + \tag{34}$$

$$\sum_{\substack{r, \\ t'+rctimep(ar)=t \\ Y(art)\notin C}} Y_{art'} + \sum_{\substack{r, \\ t'+rctimep(ar)=t \\ Y(art)\in C}} Y_{art'}$$

Similarly, equation (25) becomes:

$$HP_{akt} = HP_{ak,t-1} - \sum_{\substack{r, \\ Y(art)\notin C}} Y_{art} - \sum_{\substack{r, \\ Y(art)\in C}} Y_{art} + \tag{35}$$

$$\sum_{\substack{u,r, \\ t'+rctimep(ar)=t \\ X(uart)\notin C}} X_{uart'} + \sum_{\substack{u,r, \\ t'+rctimep(ar)=t \\ X(uart)\in C}} X_{uart'}$$

Since cascade $L-1$ produces a feasible solution (say $\overline{X}, \overline{Y}, \overline{H}, \overline{HP}$) for $T \in \{TCAS_{L-1} \cap TCAS_L\}$, we can combine (34), (35), and (28) into (27), separating elements of C:

$$cumsup_{at} \geq \tag{36}$$

$$\left(\sum_{\substack{u,r, \\ (t-m)\leq t'\leq t \\ \overline{X}(uart')\notin C}} mtime_{art't} \cdot \overline{X}_{uart'} + \sum_{\substack{u,r, \\ (t-m)\leq t'\leq t \\ \overline{X}(uart')\in C}} mtime_{art't} \cdot \overline{X}_{uart'} \right) +$$

$$\left(\sum_{\substack{r, \\ (t-m)\leq t'\leq t \\ \overline{Y}(art')\notin C}} mtime_{art't} \cdot \overline{Y}_{art'} + \sum_{\substack{r, \\ (t-m)\leq t'\leq t \\ \overline{Y}(art')\in C}} mtime_{art't} \cdot \overline{Y}_{art'} \right) +$$

$$\left(\sum_i \overline{H}_{ai,t-1} - \sum_{\substack{u,r, \\ \overline{X}(uart)\notin C}} \overline{X}_{uart} - \sum_{\substack{u,r, \\ \overline{X}(uart)\in C}} \overline{X}_{uart} + \sum_{\substack{r, \\ t'+rctimep(ar)=t \\ \overline{Y}(art)\notin C}} \overline{Y}_{art'} + \sum_{\substack{r, \\ t'+rctimep(ar)=t \\ \overline{Y}(art)\in C}} \overline{Y}_{art'} \right) +$$

$$\left(\sum_k \overline{HP}_{ak,t-1} - \sum_{\substack{r, \\ \overline{Y}(art)\notin C}} \overline{Y}_{art} - \sum_{\substack{r, \\ \overline{Y}(art)\in C}} \overline{Y}_{art} + \sum_{\substack{u,r, \\ t'+rctimep(ar)=t \\ \overline{X}(uart)\notin C}} \overline{X}_{uart'} + \sum_{\substack{u,r, \\ t'+rctimep(ar)=t \\ \overline{X}(uart)\in C}} \overline{X}_{uart'} \right)$$

Since cancelled missions use the full period t, $\quad mtime_{art't} = 1$ for $\overline{X}_{uart}, \overline{Y}_{art} \in C$.

Thus, (36) can be simplified by cancellation:

$$\sum_{\substack{u,r, \\ (t-m)\le t'\le t \\ \overline{X}(uart')\notin C}} mtime_{art't} \cdot \overline{X}_{uart'} + \sum_{\substack{r, \\ (t-m)\le t'\le t \\ \overline{Y}(art')\notin C}} mtime_{art't} \cdot \overline{Y}_{art'} +$$

$$\left(\sum_i \overline{H}_{ai,t-1} - \sum_{\substack{u,r, \\ \overline{X}(uart)\notin C}} \overline{X}_{uart} + \sum_{\substack{r, \\ t'+rctimep(ar)=t \\ \overline{Y}(art)\notin C}} \overline{Y}_{art'} + \sum_{\substack{r, \\ t'+rctimep(ar)=t \\ \overline{Y}(art)\in C}} \overline{Y}_{art'} \right) +$$

$$\left(\sum_k \overline{HP}_{ak,t-1} - \sum_{\substack{r, \\ \overline{Y}(art)\notin C}} \overline{Y}_{art} + \sum_{\substack{u,r, \\ t'+rctimep(ar)=t \\ \overline{X}(uart)\notin C}} \overline{X}_{uart'} + \sum_{\substack{u,r, \\ t'+rctimep(ar)=t \\ \overline{X}(uart)\in C}} \overline{X}_{uart'} \right)$$

which is greater than or equal to:

$$\ge \sum_{\substack{u,r, \\ (t-m)\le t'\le t \\ \overline{X}(uart')\notin C}} mtime_{art't} \cdot \overline{X}_{uart'} + \sum_{\substack{r, \\ (t-m)\le t'\le t \\ \overline{Y}(art')\notin C}} mtime_{art't} \cdot \overline{Y}_{art'} +$$

$$\left(\sum_i \overline{H}_{ai,t-1} - \sum_{\substack{u,r, \\ \overline{X}(uart)\notin C}} \overline{X}_{uart} + \sum_{\substack{r, \\ t'+rctimep(ar)=t \\ \overline{Y}(art)\notin C}} \overline{Y}_{art'} \right) +$$

$$\left(\sum_k \overline{HP}_{ak,t-1} - \sum_{\substack{r, \\ \overline{Y}(art)\notin C}} \overline{Y}_{art} + \sum_{\substack{u,r, \\ t'+rctimep(ar)=t \\ \overline{X}(uart)\notin C}} \overline{X}_{uart'} \right)$$

By (24) and (25), the parenthetical terms above equal the period t inventory under the modified null solution, say $\widetilde{H}, \widetilde{HP}$. Along with the modified null's mission variables, $\widetilde{X}, \widetilde{Y}$, the above equals:

$$= \sum_{\substack{u,r, \\ (t-m)\leq t' \leq t}} mtime_{art't} \cdot \widetilde{X}_{uart'} + \sum_{\substack{r, \\ (t-m)\leq t' \leq t}} mtime_{art't} \cdot \widetilde{Y}_{art'} + \sum_i \widetilde{H}_{ait} + \sum_k \widetilde{HP}_{akt}$$

which is less than or equal to $cumsup_{at}$, demonstrating feasibility of (27) using the modified null solution.

While THRUPUT II always has a feasible solution using the temporal cascades heuristic, this does not generalize well, and must be considered for every model where the temporal cascades heuristic is employed.

When the temporal cascades heuristic was implemented in GAMS on a large THRUPUT II data set (dual major regional contingency), an IBM RS6000 allowed about 25 periods per cascade without core dumping. However, that was without incorporating any of the streamlining changes suggested above. Those changes will allow considerable lengthening of each cascade, or permit an increase in model fidelity through less aggressive unit and airfield aggregation.

GAMS implementation of this technique to date was somewhat cumbersome. The current cascade's results were written to a file, which was then read by the next cascade. The cascade loop was conducted by a script file executed by the operating system. This technique ensured GAMS freed the memory of variables once they had been aggregated– a difficulty encountered when running a large data set. The method proposed by GAMS Development Corporation is to write the variable levels not needed in subsequent cascades to a file, then release those variables' memory for the next cascade. This sparsely documented feature allows the cascades to be run within a GAMS loop statement, which is faster and more reliable. Another potential advantage of this method lies in the computational savings derived from both upper and lower bound problems using similar bases. If these improvements still prove

unwieldy, the method could be implemented without GAMS, which will speed the solution, but reduce its attractiveness to AFSAA.

CASCADING LOWER BOUND

This section considers four principle strategies for computing the lower bound of a model with the structure described: 1) decomposition; 2) demand relaxation; 3) temporal relaxation; and 4) aggregation. Below is an outline of these ideas.

Decomposition

At first, the notion of decomposition is appealing because of structural characteristics of the model. Benders' decomposition could be performed using fleets of aircraft as subproblems, with delivery requirements and MOG constraints providing the cuts. Although aircraft fleets provide good subproblems from the standpoint of model structure, they unfortunately still require LP, rather than network solution algorithms. Because of the large number of side constraints, it appears unlikely that network algorithms can be exploited in the Air Force problem. Also, this approach forsakes any computations made by the temporal cascades heuristic. Additionally, the amount of memory consumed by the cuts becomes very large, which could lead to the core dumping already plaguing the model. Finally, the MOG constraints appear to be the most binding for typical instances of the model. Thus, convergence could be very slow, as the hardest decisions are "in the cuts." Consequently, Benders' decomposition is probably not the most effective method for this problem..

Demand Relaxation

Since the THRUPUT II objective function consists almost entirely of penalty terms from the extended staircase constraints, reducing the number of demand constraints

(the number of units requiring movement) is a model relaxation. Solving several such problems which together consider all units exactly once is also a relaxation, since only a subset of units "compete" for infrastructure resources in each problem. A careful partition of units (one that groups units which tend to compete for the same infrastructure resources) should result in a fair lower bound, and one that can be strengthened by Lagrangian multiplier search. Lagrangian relaxation has a lot of merit, since the upper bound method (temporal cascades) provides the required multiplier search. For demand relaxation, the relaxed constraints consist of the aircraft and (perhaps) the airfield constraints, which are of the staircase and single period variety. Unfortunately, there are many aircraft capacity constraints. Additionally, the airfield constraints tend to be the most binding. Consequently, a formulation which avoids relaxing *all* of the infrastructure constraints should produce better results. The next method pursues such a scheme; it relaxes none of the single period constraints and only a subset of the staircase constraints. In return, it relaxes the extended staircase constraints, which are left intact in the demand relaxation scheme.

Time Based, or Lagrangian Cascade

In order to incorporate the temporal nature of the upper bound heuristic, the model's time periods can be partitioned into subproblems. In turn, these subproblems can be solved by relaxing constraints that include variables from overlapping time periods. The structure of the problem lends some advantage to this technique, since most constraints have a minimal overlap. Additionally, the constraints with large overlaps (extended staircase) have bounded duals, so the associated penalties in the relaxed problem will stay within reasonable limits. Most importantly, however, is the availability of excellent dual variables from the associated temporal cascade heuristic. These are very compelling reasons to consider this technique. Of course, the duals from the temporal cascades are available to both relaxation methods, but the demand

method relaxes many more constraints than the time based method.

Despite the advantages, Lagrangian relaxation across time is not without difficulty. Foremost is the issue of the "most binding" constraints, which tend to be the staircase type in THRUPUT II (specifically the MOG constraints). Hopefully, the price accuracy (due to the accuracy of the temporal cascades) and minimal overlap will ameliorate this drawback. In contrast, most every extended staircase (demand) constraint has to be relaxed, since they frequently "cover" 20 or more time periods. Another difficulty lies in the block constraints, which may not be conveniently partitioned into the Lagrangian subproblems. This is a small concern in the Air Force problem, since there is no consensus as to how to model aircraft utilization limits. Finally, there is not yet a method to feedback information gleaned from the lower bound back into the upper bound problem, should the user desire a second iteration to tighten the bounds. Fortunately, knowledge of the most "violated" extended staircase constraints may provide an excellent basis to "tune" the penalties in the next temporal cascade iteration. There are numerous possibilities.

Below is a schematic, notation and formulation for the proposed method of Lagrangian relaxed cascade:
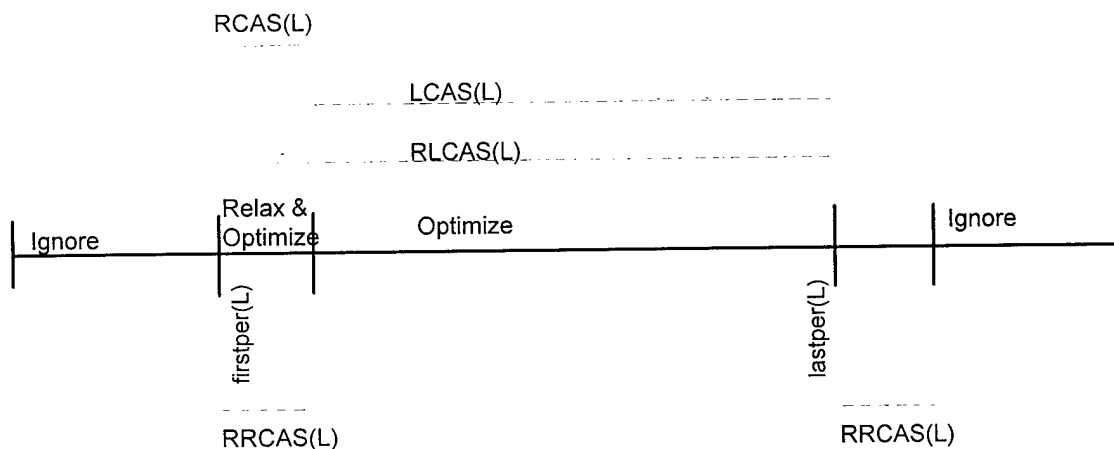
Lagrangian Cascade Notation.—

$firstper_L$ First period in cascade L

$lastper_L$ Last period in cascade L

$RCAS_L = \{t : firstper_L \leq t < (firstper_L + m), L \neq 1\}$ The overlap period at the beginning of the current cascade

$LCAS_L = \{t : \max(t \in RCAS_L) < t \leq lastper_L\}$ The non-overlapping period of the current cascade

$RLCAS_L = \{RCAS_L \cup LCAS_L\}$ All time periods solved for in the current cascade

$RRCAS_L = \{RCAS_L \cup \{t : lastper_L < t \leq lastper_L + m\}\}$ $RCAS$ plus the succeeding overlap periods. These are included since Lagrange multipliers from these periods will be applied against $LCAS$ variables

$UL_L = \{u : UT_u \cap RLCAS_L \neq \emptyset, UT_u \cap RLCAS_{L+1} = \emptyset\}$ Partition of U into the cascades L. This scheme places u into the last cascade in u's delivery window

Summary of Lagrangian Cascade Algorithm.—

0) Set $L = 1, \mid RLCAS \mid = \mid TCAS \mid - m$ (from temporal cascades model). Set $firstper_1 = 1$, $lastper_1 = \mid RLCAS \mid$. Establish other sets as described in notation.

1) Solve Lagrangian cascade subproblem using the dual variables α_u, β_t from the associated temporal cascades solution.

2) If $lastper_L \geq \mid T \mid$, terminate and report sum of objective values across all iterations as the lower bound. Otherwise, set $firstper_{L+1} = firstper_L + \mid RLCAS \mid$, and $lastper_{L+1} = lastper_L + \mid RLCAS \mid$.

3) Set $L = L + 1$, update sets $RCAS_L, LCAS_L, RLCAS_L, RRCAS_L$.

4) Return to 1.

Lagrangian Cascade Formulation.—

38

$$\underline{Z} = \min \sum_u P_u +$$

$$\sum_u \alpha_u \left(d_u - \sum_t X_{ut} - P_u \right) + \sum_{t \in (\cup_L RCAS(L))} \beta_t \left(s_t - \sum_u f_u \cdot X_{u,t-m} - \sum_u f_u \cdot X_{ut} \right)$$

OBJECTIVE: Minimize the sum of nondelivery penalties, plus Lagrangian penalties. Here α_u is the extended staircase dual variable associated with the last cascade overlapping of u's delivery window in the temporal cascades' solution. β_t is the staircase dual variable associated with the corresponding time period of the temporal cascades' solution.

$$\text{ST} \qquad \sum_u f_u \cdot X_{u,t-m} + \sum_u f_u \cdot X_{ut} \le s_t \qquad \forall t \in \cup_L LCAS_L \qquad (\beta_t)$$

STAIRCASE: Enforce for the constraints which do not overlap a relaxed window ($RCAS$).

$$\sum_u a_u \cdot X_{ut} \le g(s_t) \qquad \forall t$$

SINGLE PERIOD: These never overlap.

$$\sum_{u,t \in B} X_{ut} \le h\left(\sum s_t \right) \qquad \forall B$$

BLOCK: Enforce all block constraints. This assumes the block constraints can be partitioned in such a manner.

This formulation is decomposed into $\mid L \mid$ distinct subproblems with $\underline{Z} = \sum_L \underline{Z}$. Note α_u is bounded by the coefficient on P_u (1), thus the P_u term is not favorable, and will remain at 0. It is left in the formulation for completeness:

$$\underline{Z}_L = \min \sum_{u \in UL(L)} P_u \cdot (1 - \alpha_u) + \sum_{u \in UL(L)} \alpha_u \cdot d_u - \sum_{\substack{u, \\ t \in RLCAS(L)}} \alpha_u \cdot X_{u,t}$$

$$+ \sum_{t \in RCAS(L)} \beta_t \cdot s_t - \sum_{\substack{u, \\ t \in RRCAS(L), \\ (t-m) \in RLCAS(L)}} \beta_t \cdot f_u \cdot X_{u,t-m} - \sum_{\substack{u, \\ t \in RCAS(L)}} \beta_t \cdot f_u \cdot X_{u,t}$$

$$\text{ST} \quad \sum_u f_u \cdot X_{u,(t-m)} + \sum_u f_u \cdot X_{ut} \leq s_t \quad \forall t \in LCAS_L \quad (\beta_t)$$

$$\sum_u a_u \cdot X_{ut} \leq g(s_t) \quad \forall t \in RLCAS_L$$

$$\sum_{u,t \in B} X_{ut} \leq h(\sum s_t) \quad \forall B \subseteq RLCAS_L$$

Lagrangian Cascades Formulation of THRUPUT II.—

Implementation of THRUPUT II by Lagrangian cascade is straightforward, but notationally cumbersome. The only problematic constraint is utilization rate enforcement, which is addressed below. The objective function for the Lagrangian formulation is:

$$\underline{Z} = \min \sum_{u,a,r,t} (latepenue_u \cdot dayslate_{uart} + TONSUE_{uart} \tag{37}$$

$$+ latepenpax_u \cdot dayslate_{uart} \cdot TPAX_{uart})$$

$$\sum_u (nogopenue \cdot UENOGO_u + nogopenpax_u \cdot PAXNOGO_u)$$

$$+ \sum_{a,t} preserve_a \cdot CONSUME_{at}$$

$$+ \sum_u \alpha_{bulk,u} (\sum_{i,k} moveue_{uik} - \sum_{a,r,t} TONSUE_{uart} - UENOGO_u)$$

40

$$+ \sum_u \alpha_{out,u}(proout_u \cdot \sum_{i,k} moveue_{uik} - \sum_{r,t,a \in A(out)} TONSUE_{uart} - UENOGO_u)$$

$$+ \sum_u \alpha_{ovr,u} \left((proover_u + proout_u) \cdot \sum_{i,k} moveue_{uik} - \sum_{\substack{r,t \\ a \in A(ovr)}} TONSUE_{uart} - UENOGO_u \right)$$

$$+ \sum_u \alpha_{pax,u}(\sum_{i,k} movepax_{uik} - \sum_{art} TPAX_{uart} - PAXNOGO_u)$$

$$+ \sum_{a,i,t \in (\cup_L RCAS(L))} \beta_{of,ait}(H_{ai,t-1} + ALLOT_{ait} + \sum_{\substack{r, \\ t'+rctime(ar)=t}} Y_{art'} - \sum_{u,r} X_{uart} - H_{ait})$$

$$+ \sum_{a,k,t \in (\cup_L RCAS(L))} \beta_{df,akt}(HP_{ak,t-1} + \sum_{\substack{u,r, \\ t'+rctime(ar)=t}} X_{uart'} - \sum_r Y_{art} - HP_{akt})$$

$$+ \sum_{a,t \in (\cup_L RCAS(L))} \beta_{pc,at}(CONSUME_{at} - \sum_{\substack{u,r, \\ (t-m) \leq t' \leq t}} mtime_{art't} \cdot X_{uart'}$$

$$- \sum_{\substack{r, \\ (t-m) \leq t' \leq t}} mtime_{art't} \cdot Y_{art'} - \sum_k HP_{akt})$$

$$+ \sum_{f,t \in (\cup_L RCAS(L))} \beta_{MOG,ft}(mogeff_f \cdot mogcap_{ft} - \sum_{\substack{u,a,r, \\ t'+rctime(afr)=t}} mogreq_{af} \cdot gtime_{afr} \cdot X_{uart'}$$

$$- \sum_{\substack{a,r, \\ t'+rctime(afr)=t}} mogreq_{af} \cdot gtime_{afr} \cdot Y_{art'})$$

where $\alpha_{bulk,u}, \alpha_{out,u}, \alpha_{ovr,u}$ are the dual variables associated with the temporal cascades solution of the *DELIVERY, OUTSIZE DELIVERY*, and *OVERSIZE DELIVERY* constraints, respectively. Similarly, $\beta_{of,ait}, \beta_{df,akt}, \beta_{pc,at}, \beta_{MOG,ft}$ are the dual variables associated with the temporal cascades solution of the *ORIGIN FLOW, DESTINATION FLOW, PLANES CONSUMED*, and *AIRFIELD CAPACITY* constraints, respectively.

This objective can be decomposed and rewritten into $|L|$ distinct subproblems with $\underline{Z} = \sum_L \underline{Z}_L$. Together with the appropriate constraints, the formulation becomes:

$$\underline{Z}_L = \min \sum_{a,t \in RLCAS(L)} preserve_a \cdot CONSUME_{at} + \tag{38}$$

41

$$\sum_{\substack{u\in UL(L)}} [\sum_{i,k} moveue_{uik}(\alpha_{bulk,u} + \alpha_{out,u} \cdot proout_u + \alpha_{ovr,u}(proout_u + proover_u))$$

$$+ \sum_{i,k} movepax_{uik} \cdot \alpha_{pax,u} + UENOGO_u(nogopenue_u - \alpha_{bulk,u} - \alpha_{out,u} - \alpha_{ovr,u})$$

$$+ PAXNOGO_u(nogopenpax_u - \alpha_{pax,u})] +$$

$$\sum_{\substack{u,r,a\in A(bulk),\\ t\in RLCAS(L)}} TONSUE_{uart}(latepenue_u \cdot dayslate_{uart} - \alpha_{bulk,u}) - \sum_{\substack{u,r,a\in A(out),\\ t\in RLCAS(L)}} TONSUE_{uart} \cdot \alpha_{out,u}$$

$$- \sum_{\substack{u,r,a\in A(ovr),\\ t\in RLCAS(L)}} TONSUE_{uart} \cdot \alpha_{ovr,u} + \sum_{\substack{u,a,r,\\ t\in RLCAS(L)}} TPAX_{uart}(latepenpax_u \cdot dayslate_{uart} - \alpha_{pax,u})$$

$$+ \sum_{\substack{a,i,t\in RRCAS(L),\\ (t-1)\in RLCAS(L)}} \beta_{of,ait} \cdot H_{ai,t-1} + \sum_{\substack{a,i,\\ t\in RCAS(L)}} \beta_{of,ait}(ALLOT_{ait} - \sum_{u,r} X_{uart} - H_{ait})$$

$$\sum_{\substack{a,i,r\\ t\in RRCAS(L),\\ t'\in RLCAS(L),\\ t'+rctimep(ar)=t}} \beta_{of,ait} \cdot Y_{art'} + \sum_{\substack{a,k,\\ t\in RRCAS(L),\\ (t-1)\in RLCAS(L)}} \beta_{df,akt} \cdot HP_{ak,t-1} - \sum_{\substack{a,k,\\ t\in RCAS(L)}} \beta_{df,akt}(\sum_r Y_{art} + HP_{akt})$$

$$+ \sum_{\substack{a,k,\\ t\in RRCAS(L),\\ t'\in RLCAS(L),\\ t'+rctimep(ar)=t}} \beta_{df,akt} \cdot \sum_{u,r} X_{uart'} + \sum_{\substack{a,t\in RCAS(L)}} \beta_{pc,at}(CONSUME_{at} - \sum_k HP_{akt})$$

$$- \sum_{\substack{u,a,r,\\ t\in RRCAS(L),\\ t'\in RLCAS(L),\\ (t-m)\le t'\le t}} \beta_{pc,at} \cdot mtime_{art't} \cdot X_{uart'} - \sum_{\substack{r,\\ t\in RRCAS(L),\\ t'\in RLCAS(L),\\ (t-m)\le t'\le t}} \beta_{pc,at} \cdot mtime_{art't} \cdot Y_{art'}$$

$$+ \sum_{\substack{f,t\in RCAS(L)}} \beta_{MOG,ft} \cdot mogeff_f \cdot mogcap_{ft} - \sum_{\substack{u,a,r,f\\ t\in RRCAS(L),\\ t'\in RLCAS(L),\\ t'+rctime(afr)=t}} \beta_{MOG,ft} \cdot mogreq_{af} \cdot gtime_{afr} \cdot X_{uart'}$$

$$- \sum_{\substack{a,r,f\\ t\in RRCAS(L),\\ t'\in RLCAS(L),\\ t'+rctime(afr)=t}} \beta_{MOG,ft} \cdot mogreq_{af} \cdot gtime_{afr} \cdot Y_{art'}$$

42

OBJECTIVE : All terms are partitioned into 1 cascade. Again, the unconstrained (except non-negativity) penalty terms will not have a favorable coefficient and will remain at 0.

Subject To (for each subproblem L)

{*all delivery constraints are relaxed*}

$$\sum_{u,r} X_{uart} + H_{ait} = H_{ai,t-1} + ALLOT_{ait} + \sum_{\substack{r, \\ t'+rctimep(ar)=t}} Y_{art'} \qquad \forall a, i, t \in LCAS_L \quad (39)$$

ORIGIN FLOW BALANCE : Original formulation of (6), but defined only for those constraints of cascade L which are not in the relaxed region, $RCAS_L$.

$$\sum_r Y_{art} + HP_{akt} = HP_{ak,t-1} + \sum_{\substack{u,r, \\ t'+rctimep(ar)=t}} X_{uart'} \quad \forall a, k, t \in LCAS_L \qquad (40)$$

DESTINATION FLOW BALANCE : Original formulation of (7), but defined only for those time periods of cascade L which are not in the relaxed region, $RCAS_L$.

$$\sum_i ALLOT_{ait} = supply_{at} \quad \forall a, t \in RLCAS_L \qquad (41)$$

AIRCRAFT ALLOTMENT : Original formulation of (8), defined only for those constraints of cascade L (none are relaxed).

$$CONSUME_{at} + \sum_i H_{ait} \le cumsup_{at} \qquad \forall a, t \in RLCAS_L \qquad (42)$$

MAXIMUM PLANES : Equation (17), defined only for those constraints of cascade L (none are relaxed).

$$\sum_{\substack{u,r,\\(t-m)\le t'\le t}} mtime_{art't} \cdot X_{uart'} + \tag{43}$$

$$\sum_{\substack{r,\\(t-m)\le t'\le t}} mtime_{art't} \cdot Y_{art'} + \sum_k HP_{akt} \le CONSUME_{at} \qquad \forall a, t \in LCAS_L$$

$PLANES\ CONSUMED$: Equation (18), but defined only for those time periods of cascade L which are not in the relaxed region, $RCAS_L$.

$$TONSUE_{uart} + paxwt \cdot TPAX_{uart} \le maxload_{ar} \cdot X_{uart} \quad \forall u, a, r, t \in RLCAS_L \tag{44}$$

$AIRCRAFT\ WEIGHT$: Original formulation of (11), defined only for those constraints of cascade L (none are relaxed).

$$paxsqft_u \cdot TPAX_{uart} + uesqft_u \cdot TONSUE_{uart} \le acsqft_a \cdot loadeff_a \cdot X_{uart} \tag{45}$$
$$\forall u, a, r, t \in RLCAS_L$$

$AIRCRAFT\ SPACE$: Original formulation of (12), defined only for those constraints of cascade L (none are relaxed).

$$TPAX_{uart} \le maxpax_a \cdot X_{uart} \quad \forall u, a, r, t \in RLCAS_L \tag{46}$$

$AIRCRAFT\ SEATS$: Original formulation of (13), defined only for those constraints of cascade L (none are relaxed).

$$\sum_{\substack{u,a,r,\\t'+rctime(afr)=t}} mogreq_{af} \cdot gtime_{afr} \cdot X_{uart'} \tag{47}$$

$$+ \sum_{\substack{a,r,\\t'+rctime(afr)=t}} mogreq_{af} \cdot gtime_{afr} \cdot Y_{art'}$$

$$\leq mogeff_f \cdot mogcap_{ft} \qquad \forall f, t \in LCAS_L$$

AIRFIELD CAPACITY LIMITIATIONS: Original formulation of (14), but defined only for those time periods of cascade L which are not in the relaxed region, $RCAS_L$.

$$\sum_{\substack{u,r, \\ (t-m)\leq t'\leq t, \\ t\in LCAS(L)}} flttime_{art't} \cdot X_{uart'} + \sum_{\substack{r, \\ (t-m)\leq t'\leq t, \\ t\in LCAS(L)}} flttime_{art't} \cdot Y_{art'} \qquad (48)$$

$$\leq \sum_{t\in LCAS(L)} urate_a \cdot CONSUME_{at} \qquad \forall a$$

AIRCRAFT UTILIZATION RATE: Enforce once per cascade, but do not include $t \in RCAS_L$, which are the relaxed time periods. This does not produce a tight enforcement, since a few periods of the iteration do not have their utilization counted. Alternatively, the constraint could be relaxed and given a Lagrangian penalty in the objective, but this is probably not warranted in this model, since utilization rate is somewhat vague by the nature of the problem.

Although the Lagrangian cascade method has not yet been implemented into a THRUPUT II model instance, it shows the most promise as a lower bounding technique. However, for completeness, there is one additional lower bound technique whose description follows.
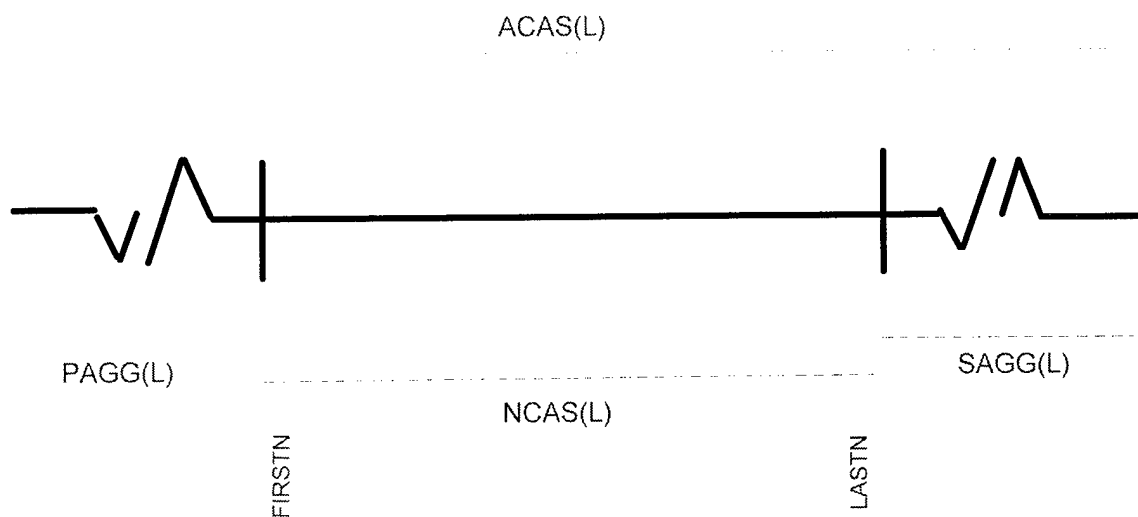
Aggregation Sandwich

The difficulties with the two methods described above (as well as others considered) motivated the consideration of aggregation as a backup to determine a lower bound. Aggregation techniques have different problems, since temporal aggregation involves column reduction as well as row reduction, which potentially violates the

rules of strict relaxation. In fact, the THRUPUT II modelling team encountered a restricted solution when pursuing a two day, versus a one day time step. The restriction occurred due to rounding of mission times in the flow balance constraints, which precluded vehicles from being relaunched until the period after the previous mission's termination. Thus, a two day time step conceivably requires more vehicle idle time, which degraded the objective. Other aspects of aggregation relaxed the model, such as the widening of unit delivery windows. So a general coarsening of the time step did not reliably produce a relaxation or restriction, just a more tractable solution with less fidelity! However, when the baseline time step is established (the time step with the minimum acceptable fidelity), a carefully constructed "sandwich" approach will hopefully produce a reasonable lower bound.

Consider 20 or so time periods in the middle of the scenario. If we add an aggregated period at either end which together incorporate the remaining time periods, we can solve, one iteration at a time, for the middle of the "sandwich." Each iteration solves for the entire scenario horizon, but only looks closely at a subset of time periods. Thus, the lower bound is not computed by summing all objective functions, but by summing a portion of each. Every unit is partitioned into a host iteration, which provides that unit's penalties for the lower bound. Summing these subsets produces a lower bound that includes each unit only once. Similarly, the vehicle consumption penalties are derived from summing the consumption over just the unaggregated periods of all the iterations.

For a lower bound, this scheme has not yet addressed the earlier concern of aggregated columns. Fortunately, some aggregations do not hurt the lower bound. If a mission is launched in the successor period of aggregation, it will consume resources only in that period, since all remaining periods are aggregated there. However, a mission launched in the predecessor period may or may not consume resources in the unaggregated period, depending on mission duration and time of launch. The

potential exists to either "overcharge" the predecessor period or "overcharge" the unaggregated periods, either of which must be avoided for a good lower bound. Assuming the predecessor aggregates more than m periods (this is always the case), the unaggregated periods cannot be charged with resources committed in the predecessor. Similarly, the predecessor resources must either: 1) be buffered with additional resources; or 2) must never be charged with more than 1 unaggregated period of resource utilization. Considering the former case, if the predecessor aggregates 20 days, it must receive 22 days of resources in order to compensate for missions which, in an unaggregated period, would optimally have launched on day 20 (and consumed resources of days 20-22). This should produce a better bound than the latter case, since most missions last longer than 1 period, and most missions cannot be launched on the last day of the predecessor aggregate. Other constraints are dealt with similarly. Below is the generic notation and formulation:

ACAS(L)

PAGG(L)   SAGG(L)

NCAS(L)

FIRSTN   LASTN

Aggregation Sandwich Notation.—

47

$firstn_L$ First unaggregated period in cascade L

$lastn_L$ Last unaggregated period in cascade L

$PAGG_L = \{t : t < firstn_L\}$. The periods prior to $firstn_L$, which have been aggregated

$PT_L = firstn_L - 1$. Predecessor aggregated time period

$SAGG_L = \{t : t > lastn_L\}$. The periods after $lastn_L$, which have been aggregated

$ST_L = lastn_L + 1$. Successor aggregated time period

$NCAS_L = \{t : firstn_L \leq t \leq lastn_L\}$. Unaggregated time periods in the current cascade

$ACAS_L = \{t : t \in PT_L \cup NCAS_L \cup ST_L\}$. Aggregated and unaggregated time periods

$UL_L = \{u : UT_u \cap NCAS_L \neq \emptyset, UT_u \cap NCAS_{L+1} = \emptyset\}$. Partition set of each unit into the last cascade in the unit's delivery window.

Summary of Aggregation Sandwich Algorithm.—

0) Establish the size of each cascade window $|NCAS|$. Set $firstn_1 = 1, lastn_1 = |NCAS|$. Set $PAGG_1 = \emptyset, SAGG_1 = t : t > lastn_1, PT_1 = 0, ST_1 = |NCAS| + 1$, $ACAS_1 = NCAS_1 \cup ST_1$.

1) Solve aggregation sandwich model.

2) Let $z_L = \sum_{u \in UL(L)} P_u$. Note this is only a portion of the total iteration objective function value.

3) If $lastn_L \geq |T|$, terminate and sum penalties computed in 2) over all iterations (since UL is a partition, each unit's penalties are counted only once). Otherwise, set $firstn_{L+1} = firstn_L + |NCAS|$, $lastn_{L+1} = lastn_L + |NCAS|$.

4) Set $L = L + 1$. Update other sets and parameters as appropriate.

5) Return to 1).

Aggregation Sandwich Formulation.—

48

$$\underline{Z} = \sum_L \underline{Z}_L \quad \text{where} \quad \underline{Z}_L = \min \sum_{u \in UL(L)} P_u$$

OBJECTIVE: yields a lower bound at algorithm termination, but where each iteration solves:

$$\min \sum_u P_u$$

which is the sum of ALL unit penalties.

$$\text{ST} \quad \sum_{t \in ACAS(L)} X_{ut} + P_u \geq d_u \quad \forall u$$

EXTENDED STAIRCASE: same as basic formulation, but sum time over the current horizon plus the aggregated periods at either end.

$$\sum_u f_u \cdot X_{ut} \leq s_t \cdot | \, PAGG_L + (m-1) \, | \quad t = PT_L$$

$$\sum_u f_u \cdot X_{u,(t-m) \in NCAS(L)} + \sum_u f_u \cdot X_{u,t} \leq s_t \quad \forall t \in NCAS_L$$

$$\sum_u f_u \cdot X_{ut} \leq s_t \cdot | \, SAGG_L + (m-1) \, | \quad t = ST_L$$

STAIRCASE: for $t \in NCAS_L$, note that we do not include any X's from PAGG. Also, the RHS in PT, ST must be increased by the amount of potential overlap. This overlap is $m-1$, since missions are launched at the beginning of a period. Thus, they spend at least 1 period on the other side of the aggregation boundary.

$$\sum_u a_u \cdot X_{ut} \leq g(s_t) \cdot | \, PAGG_L \, | \quad t = PT_L$$

$$\sum_u a_u \cdot X_{ut} \leq g(s_t) \quad \forall t \in NCAS_L$$

$$\sum_u a_u \cdot X_{ut} \leq g(s_t) \cdot | \, SAGG_L \, | \quad \forall t = ST_L$$

49

SINGLE PERIOD: Same as basic formulation.

$$\sum_u X_{ut} \leq h(\sum s_t) \cdot \mid B \subseteq PAGG_L \mid \qquad t = PT_L$$

$$\sum_{u,t \in B} X_{ut} \leq h(\sum s_t) \qquad \forall B \subseteq NCAS_L$$

$$\sum_u X_{ut} \leq h(\sum s_t) \cdot \mid B \subseteq SAGG_L \mid \qquad t = ST_L$$

BLOCK: Assume that aggregation boundaries and B are coincident.

THRUPUT Specific Aggregation Sandwich Discussion..—

Although the aggregation sandwich approach should provide a reasonable lower bound to the cascade, the formulation is tricky where aggregation transitions from disaggregation. Additionally, aggregation can overstate a unit's "deliverability" in at least two ways. Since the end of a scenario tends to be less constrained than the beginning, the successor aggregate offers lots of resources that should not be used unless a unit's delivery window extends all the way to the end. One can partially redress this by choosing each unit's "home" iteration as the one which includes the last feasible delivery day. This makes the unit ineligible for delivery variables in the successor, and looks most carefully at the period where late penalties may be incurred. The second, potentially gross relaxation involves the sparse staircase constraints with a constant RHS. MOG utility in the THRUPUT II model provides a good example. Assuming the above method for "hosting" a unit is observed, unit deliveries are often permitted in the predecessor period. Aggregation may give this unit many more airfield–days for delivery, which could be significant in a MOG constrained destination. Much of this is unavoidable, although when aggregating, MOG-days should be summed only across those periods which might have deliveries to that airfield.

Despite its drawbacks, the aggregation sandwich approach has substantial merit.

While the Lagrangian technique probably holds more promise, the sandwich could serve as a backup should the other lower bound method fail.

SUMMARY

The methods described provide a solution to the tractability difficulties we have encountered with the THRUPUT II model. The upper bound heuristic finds a feasible solution on the first pass, and early research indicates that solution to be of high quality. Moreover, the upper bound method could potentially iterate by stiffening the elastic demand penalties whose constraints are found to be tight in a later cascade. Furthermore, the user noted that solving the whole monolith provides results that are "too good," indicating that planners never have as much future knowledge as we give the mathematical abstraction. Thus, the temporal cascades heuristic not only yields a computationally tractable solution, but a more realistic one as well. Since in general, this is not the case, the heuristic must be supported with an error bound.

The bounding schemes are as yet untested, but the Lagrangian cascade method shows the most promise. It incorporates dual prices from the temporal cascades in serial fashion--which means much of an upper bound basis may be retained as a starting solution for the relaxation. Additionally, the bound could be improved by adjusting the Lagrangian penalties associated with the iterating upper bound heuristic. However, with sufficient overlap of the temporal cascades, iterating may not often be required, since the upper bound solution (and the associated duals feeding the lower bound) will be of demonstrably adequate quality.

Together, the two methods used in concert appear to form a compatible and efficient solution strategy for THRUPUT II. Further research should verify this, as well as assert a variation of the strategy on more general staircase linear programs.

REFERENCES

[1] Ahuja, R., Magnanti, R., and Orlin, J., *Network Flows*, Prentice Hall, Englewood Cliffs NJ, 1993.

[2] Aronson, J., Morton, T., and Thompson, G., "A Forward Simplex Method for Staircase Linear Programs," *Management Science* 31 (1985), 664-679.

[3] Bazaraa, M., and Sherali, H., and Shetty, C., *Nonlinear Programming*, John Wiley & Sons, New York, 1993.

[4] Benders, J., "Partitioning Procedures for Solving Mixed Variables Programming Problems," *Numerische Mathematik*, 4 (1962), 238-252.

[5] Birge, J., "Decomposition and Partitioning Methods for Multistage Stochastic Linear Programs," *Operations Research* 33 (1985), 989-1007.

[6] Dantzig, G., and Wolfe, P., "Decomposition Principle for Linear Programs," *Operations Research*, 8 (1960), 101-111.

[7] Geoffrion, A., and Graves, G., "Multicommodity Distribution System Design by Benders Decomposition," *Management Science* 20 (1974), 822-844.

[8] Brown, G., Graves, G., and Honczarenko, M., "Design and Operation of a Multi-commodity Production/Distribution System Using Primal Goal Decomposition," *Management Science* 33 (1987), 1469-1480.

[9] Brown, G., and Graves, G, and Ronen, D., "Scheduling Ocean Transportation of Crude Oil," *Management Science* 33 (1987), 335-346.

[10] Glassey, C., "Nested Decomposition and Multistage Linear Programs," *Management Science* 20 (1973), 282-292.

[11] Goggins, D., *Stochastic Modeling for Airlift Mobility*, Master's Thesis, Operations Research Department, Naval Postgraduate School, Ca., 1995.

[12] Han, S., "A Successive Projection Method," *Mathematical Programming* 40 (1988), 1-14.

[13] Hearn, D., and Lawphongpanich, S., "Lagrangian Dual Ascent by Generalized Linear Programming," *Operations Research Letters* 8 (1989), 189-196.

[14] Held, M., Wolfe, P., and Crowder, H., "Validation of Subgradient Optimization," *Mathematical Programming* 6 (1974), 62-88.

[15] Ho, J., and Manne, A. S., "Nested Decomposition for Dynamic Models," *Mathematical Programming* 6 (1974), 121-140.

[16] Jayakumar, M., and Ramasesh, R., "A Solution-Cascading Approach to the Decomposition of Staircase Linear Programs," *J Opl. Res. Soc.* 3 (1994), 301-308.

[17] Killingsworth, P., and Melody, L., "CONOP Air Mobility Optimization Model," RAND, Santa Monica, Ca., 1994.

[18] Kim, S., and Ahn, H., "Convergence of a Generalized Subgradient Method for Nondifferentiable Convex Optimization," *Mathematical Programming* 50 (1991), 75-80.

[19] Lim, T., *Strategic Airlift Assets Optimization Model*, Master's Thesis, Operations Research Department, Naval Postgraduate School, Ca., 1994.

[20] Manne, A., "Sufficient Conditions for Optimality in an Infinite Horizon Development Plan," *Econometrica* 38 (1970), 18-38.

[21] Morton, D., "An Enhanced Decomposition Algorithm for Multistage Stochastic Hydroelectric Scheduling," NPS-OR-94-001, Naval Postgraduate School, Ca. 1994.

[22] Morton, D., Rosenthal, R., and Lim, T., "Optimization Modeling for Airlift Mobility," *Military Operations Research* (to appear).

[23] Morton, T., "Forward Algorithms for Forward Thinking Managers," *Applications of Management Science Vol 1*, Schultz (Ed), JAI Press Inc., 1981.

[24] Parker, R., and Rardin, R., *Discrete Optimization*, Academic Press, Boston, 1988.

[25] Tseng, P., "Dual Coordinate Ascent Methods for Non-Strictly Convex Minimization," *Mathematical Programming* 59 (1993), 231-247.

[26] Turker, Y., *Route and Column Generation Methods for Airlift Mobility Optimization*, Master's Thesis, Operations Research Department, Naval Postgraduate School, Ca., 1995.

[27] Van Slyke, R., and Wets, R., "L-Shaped Linear Programs with Applications to Optimal Control and Stochastic Linear Programs," *SIAM J. Appl. Math.* 17 (1969), 638-663

[28] Walker, S., *Evaluating End Effects for Linear and Integer Programs Using Infinite-Horizon Linear Programming*, Ph.D. Dissertation, Operations Research Department, Naval Postgraduate School, Ca., 1995.

[29] Wing, V., Rice, R., Sherwood, R., and Rosenthal., R., "Determining the Optimal Mobility Mix," Joint Staff(J8), Force Design Division, The Pentagon, Washington, DC, 1 October 1991.

[30] Yost, K., "The THRUPUT Strategic Airlift Flow Optimization Model, "Air Force Studies and Analyses Agency, 30 June 1994.

[31] Zipkin, P., "Bounds for Row Aggregation in Linear Programming," *Operations Research* 28 (1980), 903-916.